

FIG. 1

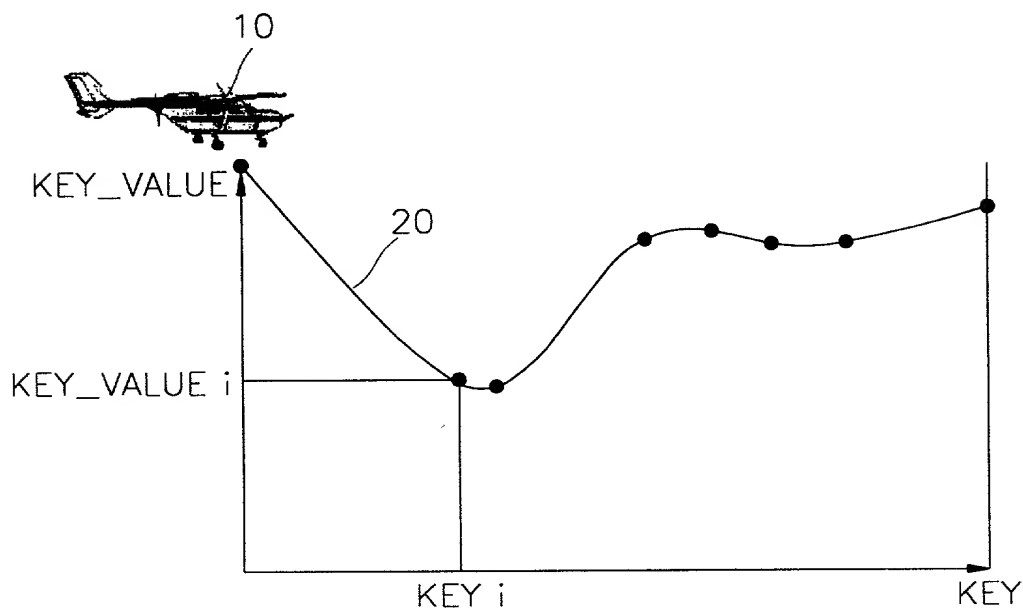


FIG. 2

```
ScalarInterpolator {
    eventin SFFloat set_fraction
    exposedField MFFloat key [ ]
    exposedField MFFloat keyValue [ ]
    eventout SFFloat value_changed
}
```

FIG. 3

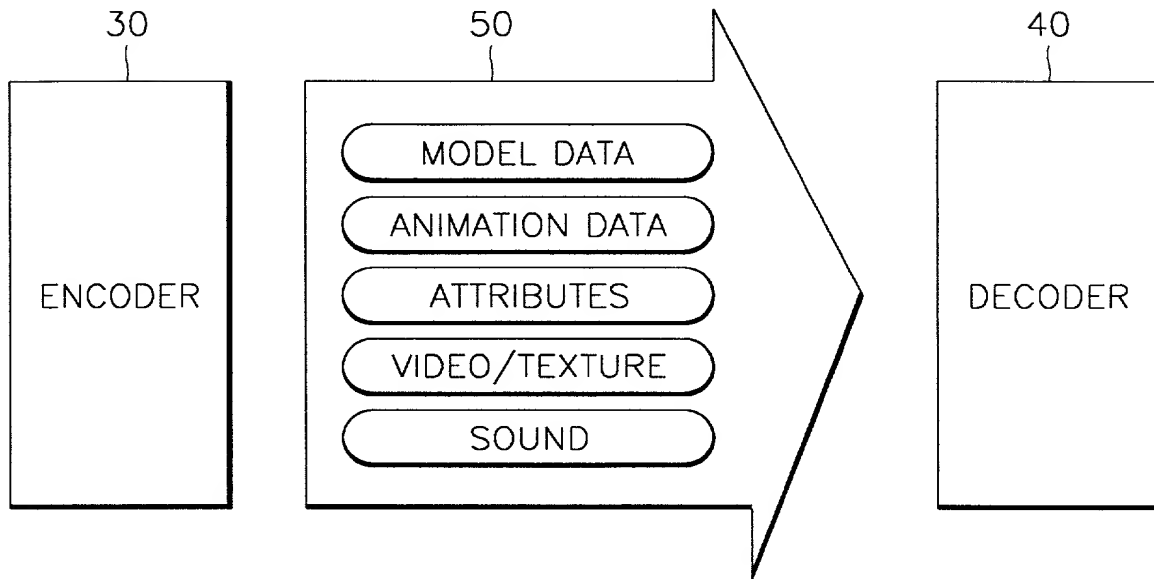


FIG. 4A

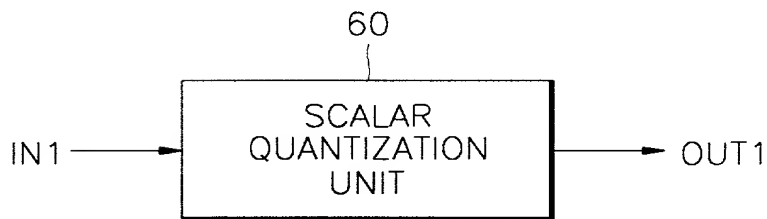


FIG. 4B

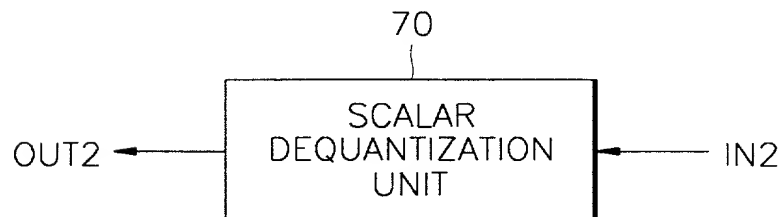


FIG. 5A

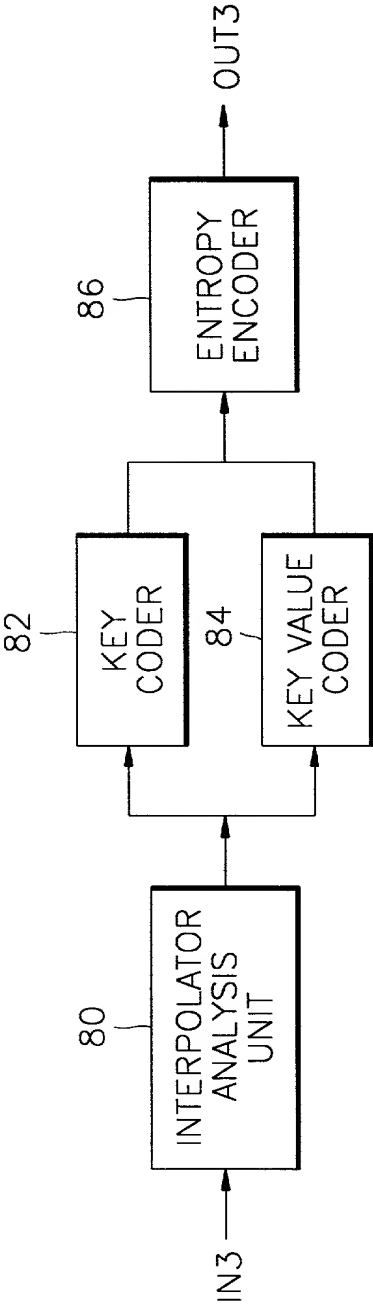


FIG. 5B

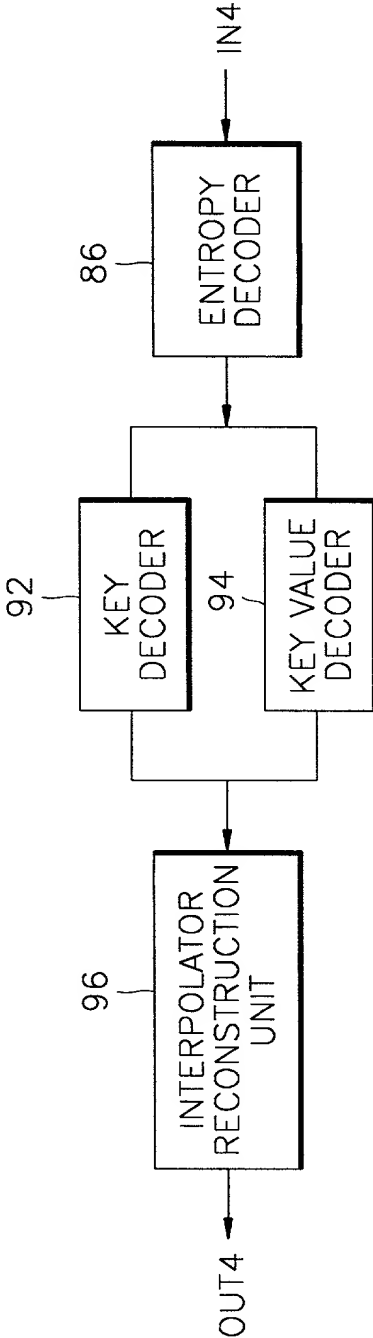


FIG. 6

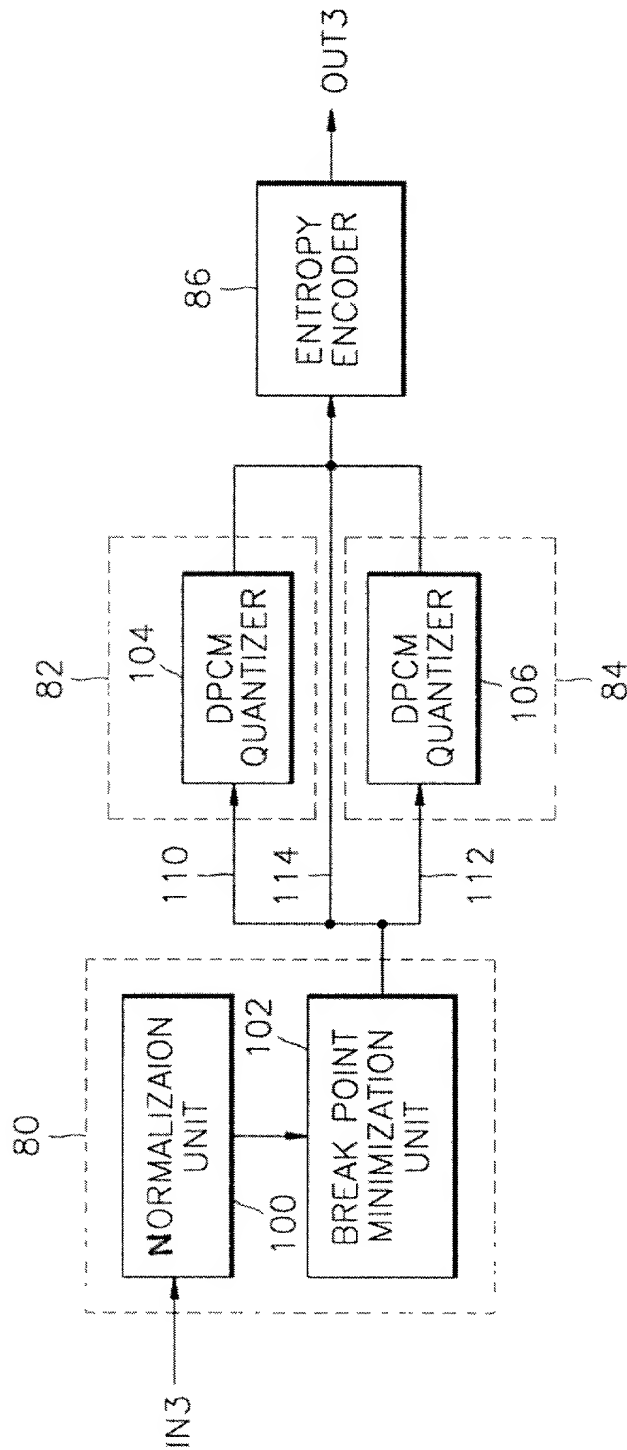


FIG. 7

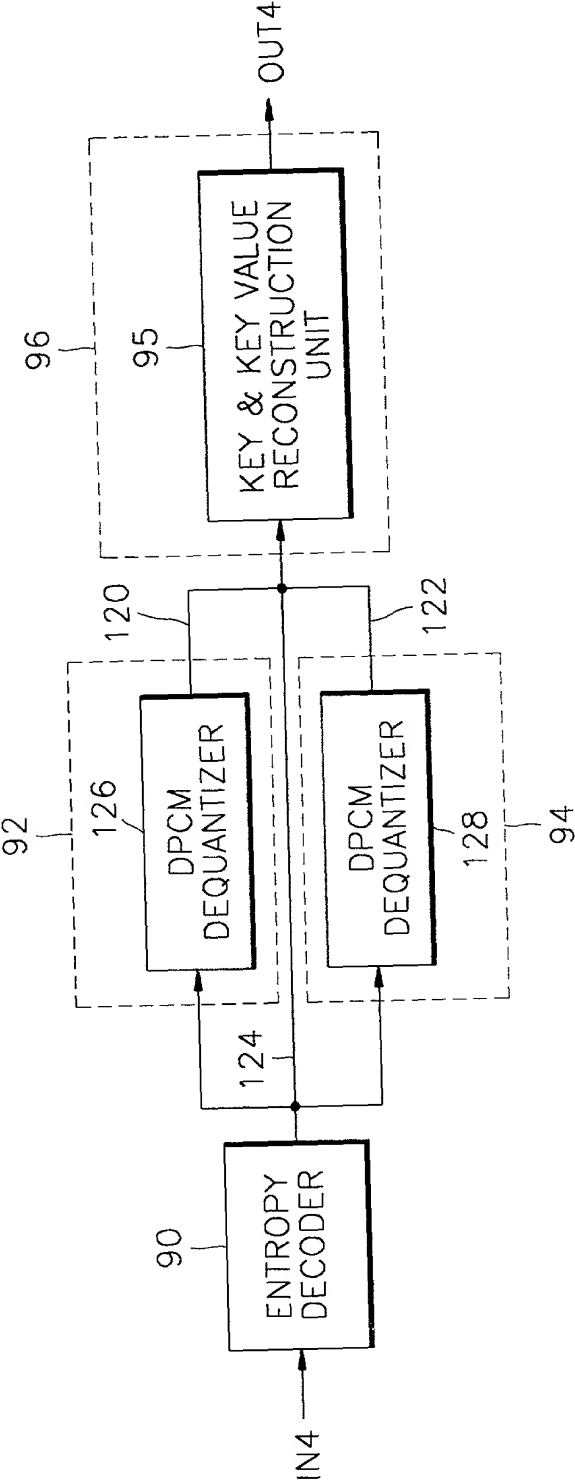


FIG. 8A

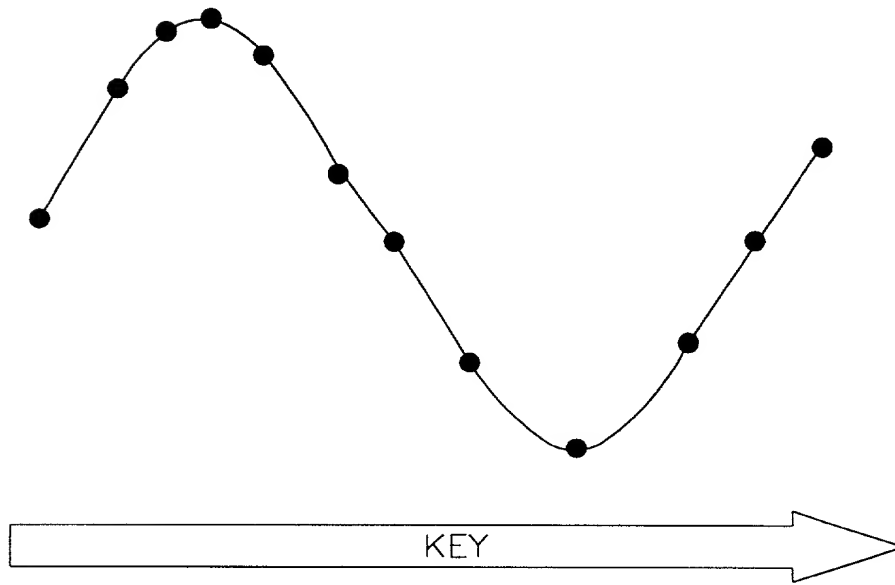


FIG. 8B

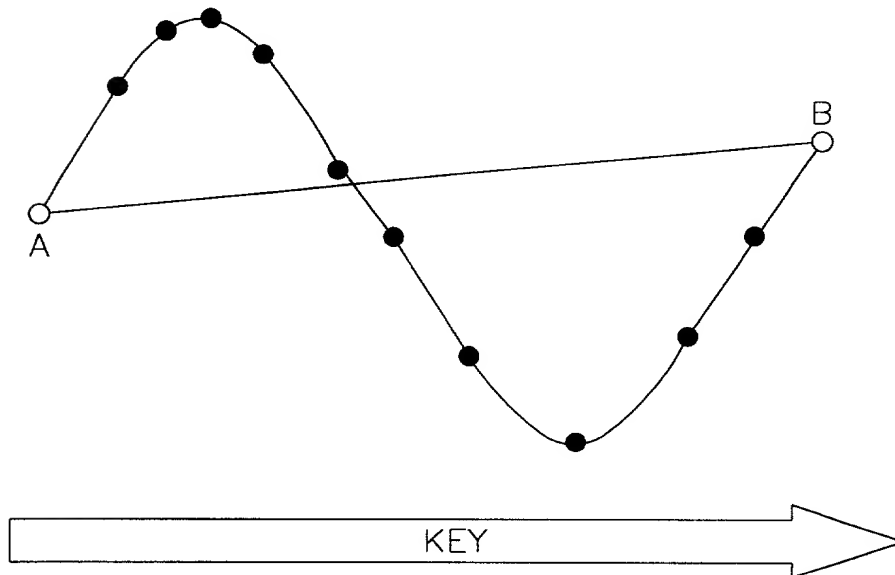


FIG. 8C

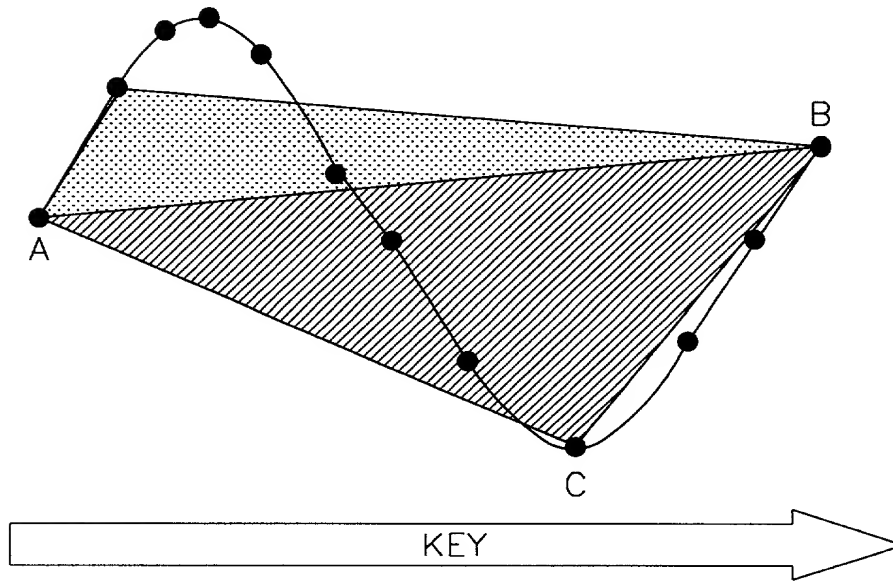


FIG. 8D

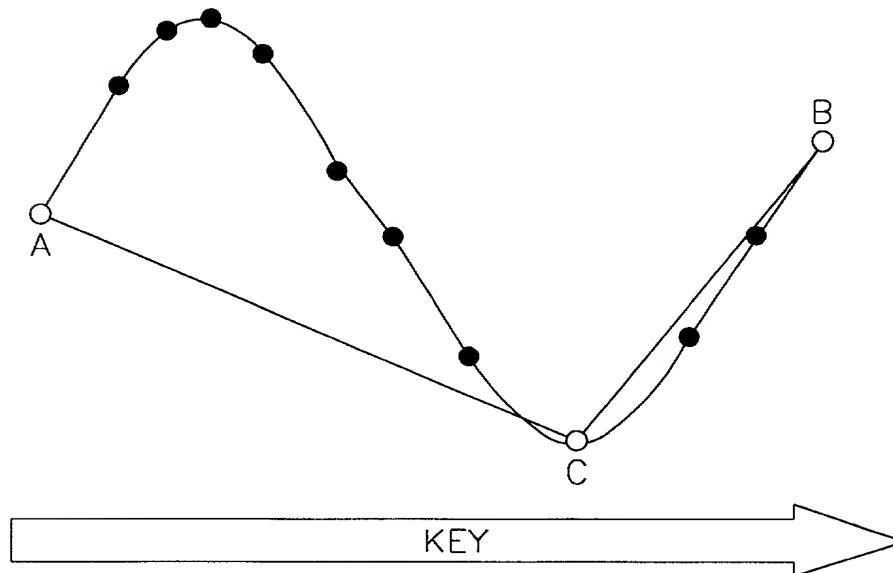


FIG. 8E

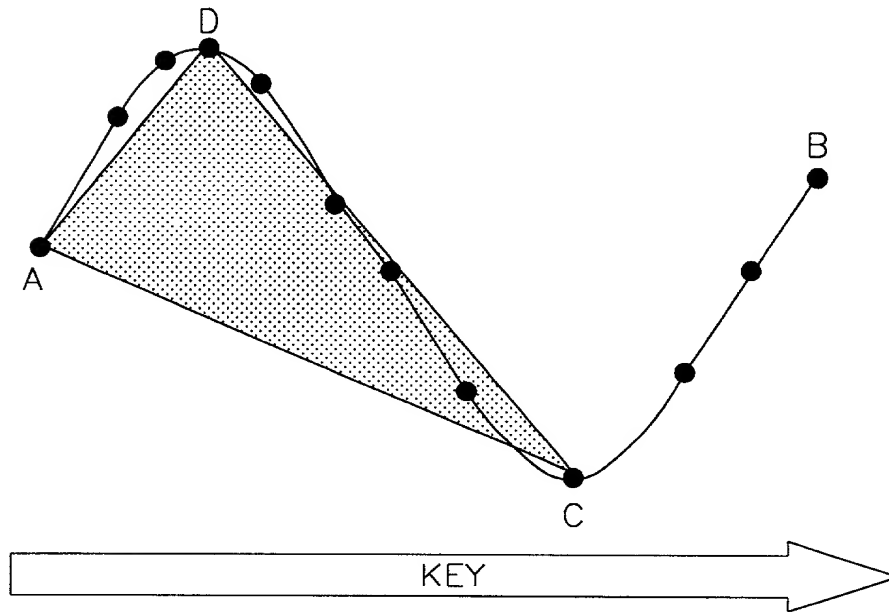


FIG. 8F

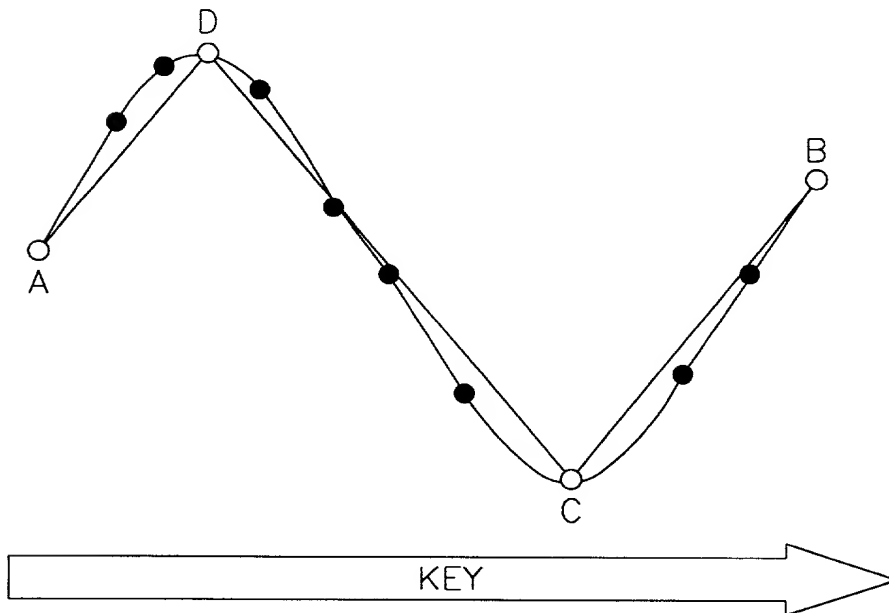


FIG. 8G

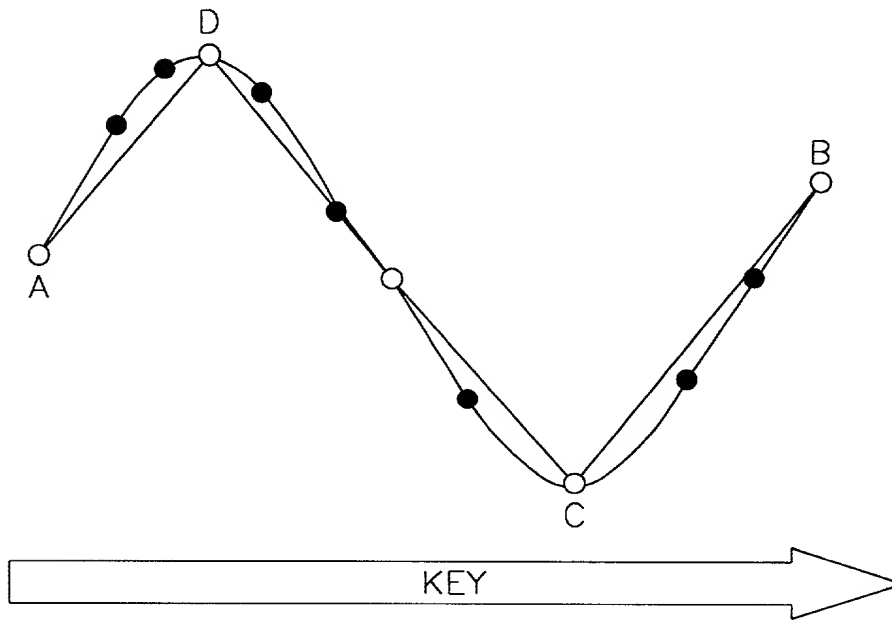


FIG. 8H

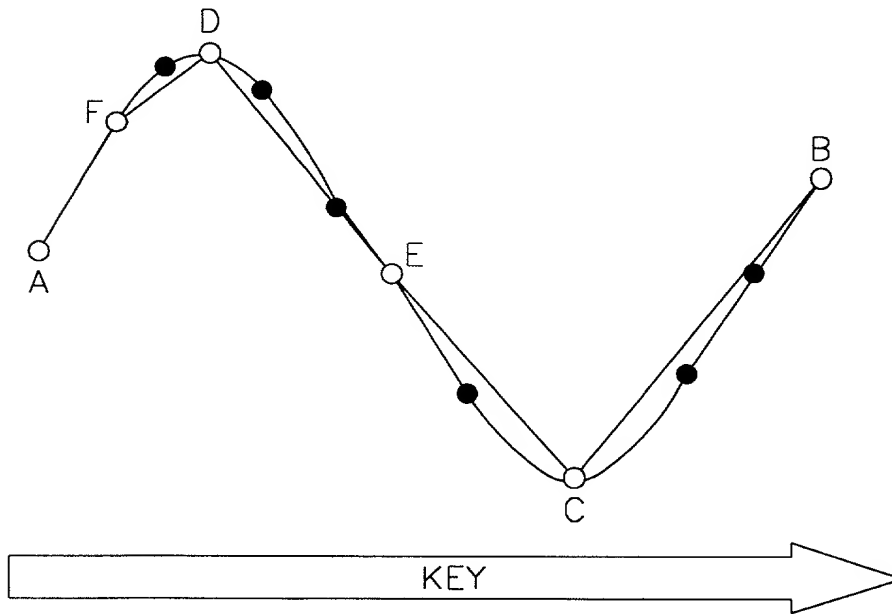


FIG. 9

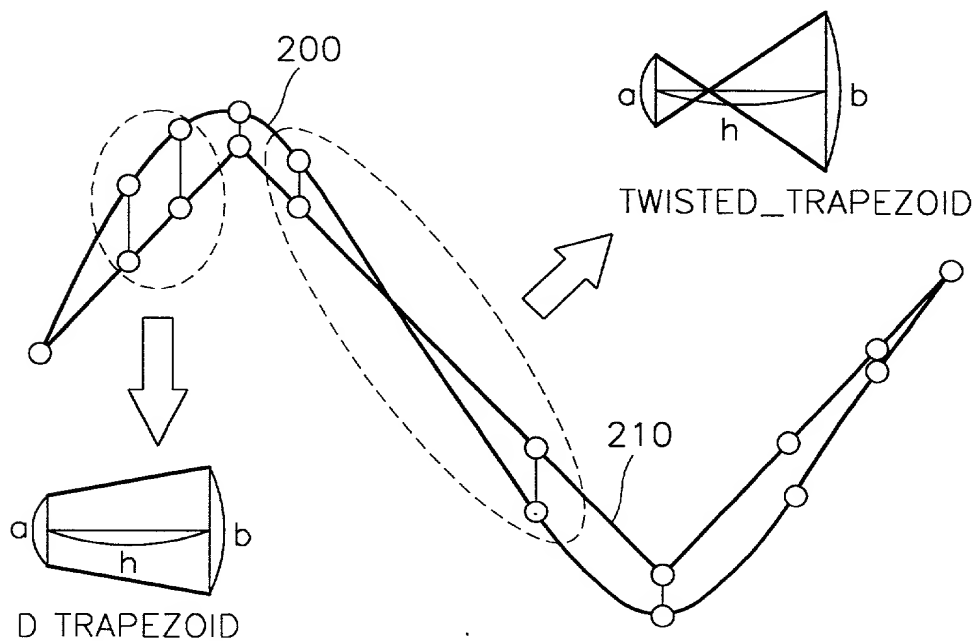


FIG. 10

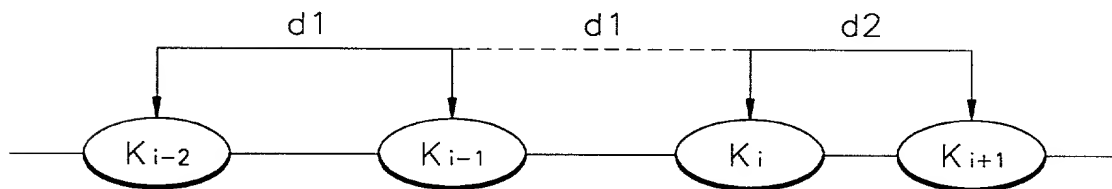


FIG. 11A

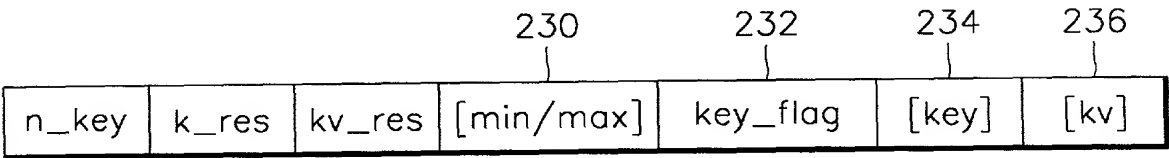


FIG. 11B

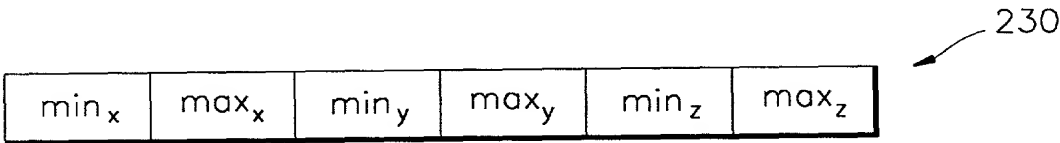


FIG. 11C

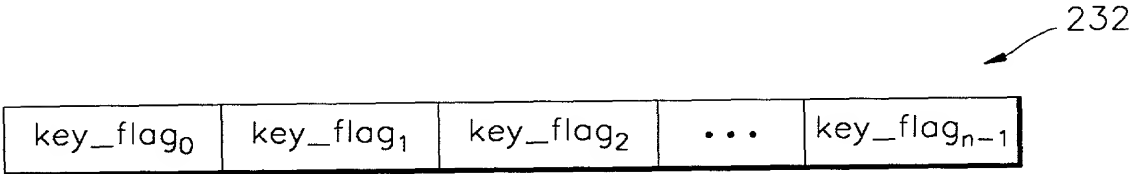


FIG. 11D

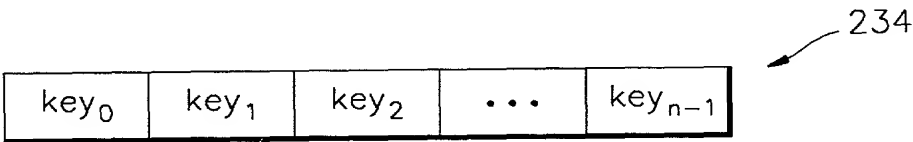


FIG. 11E

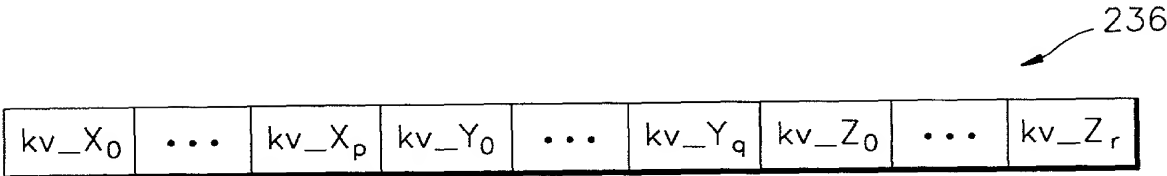


FIG. 12

CompressedPositionInterpolator ()	No. of bits	Mnemonic
{		
decodeAAC_start();		
n_key	16	uimsbf
k_res	16	uimsbf
kv_res	16	uimsbf
min_x	16	uimsbf
max_x	16	uimsbf
min_y	16	uimsbf
max_y	16	uimsbf
min_z	16	uimsbf
max_z	16	uimsbf
for (i=0; i<n_key; i++)		
{		
decodeAAC(key_flag[i], Key_flag_Context);		vlc1bf
}		
for (i=0; i<n_key; i++)		
{		
decodeAAC(key[i], Key_Context);		vlc1bf
}		
decipher_key_flag(key_flag);		
for (i=0; i<n_kv_x; i++)		
{		
decodeAAC(kv[i], Kv_Context);		vlc1bf
Inv_Quantize(kv[i]);		
}		
for (i=0; i<n_kv_y; i++)		
{		
decodeAAC(kv[i], Kv_Context);		vlc1bf
Inv_Quantize(kv[i]);		
}		
for (i=0; i<n_kv_z; i++)		
{		
decodeAAC(kv[i], Kv_Context);		vlc1bf
Inv_Quantize(kv[i]);		
}		
decodeAAC_finish();		
}		

FIG. 13A

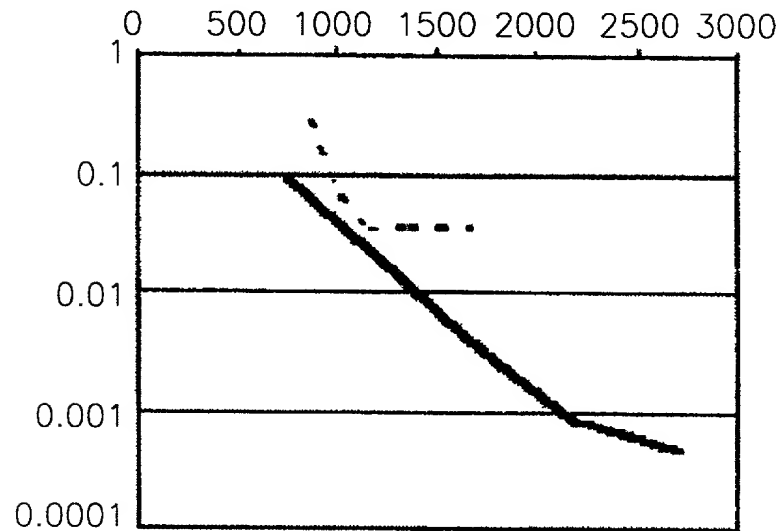


FIG. 13B

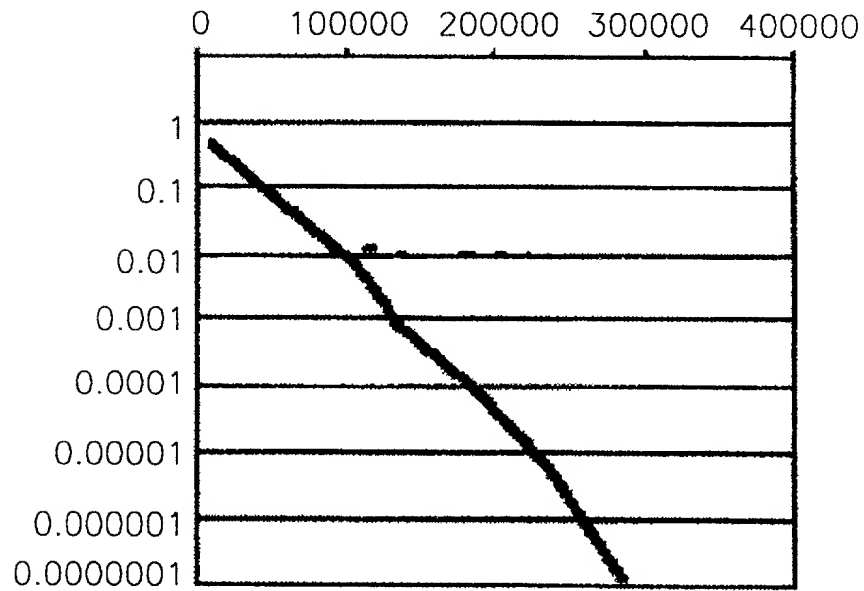


FIG. 13C

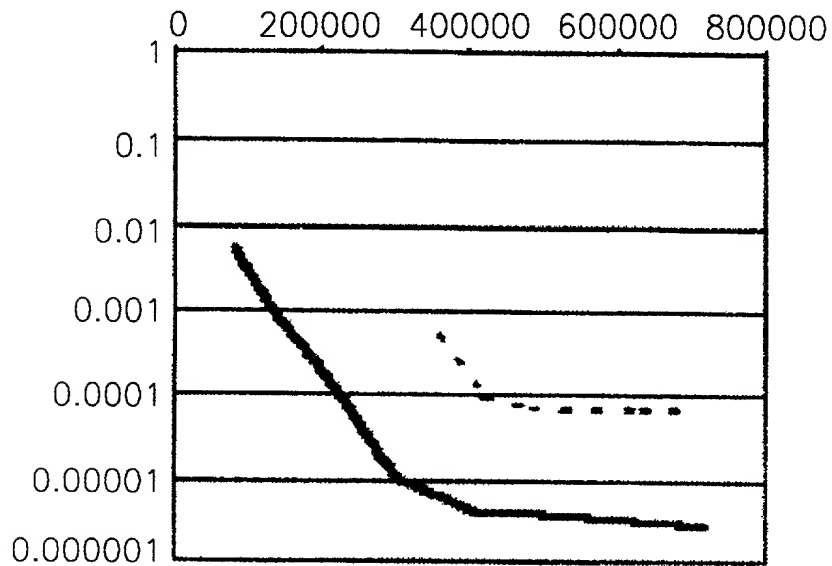


FIG. 13D

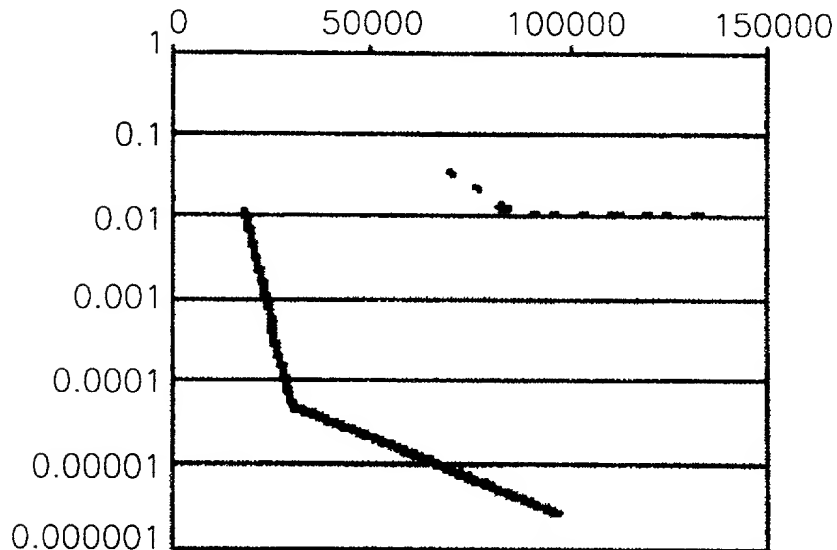


FIG. 14A

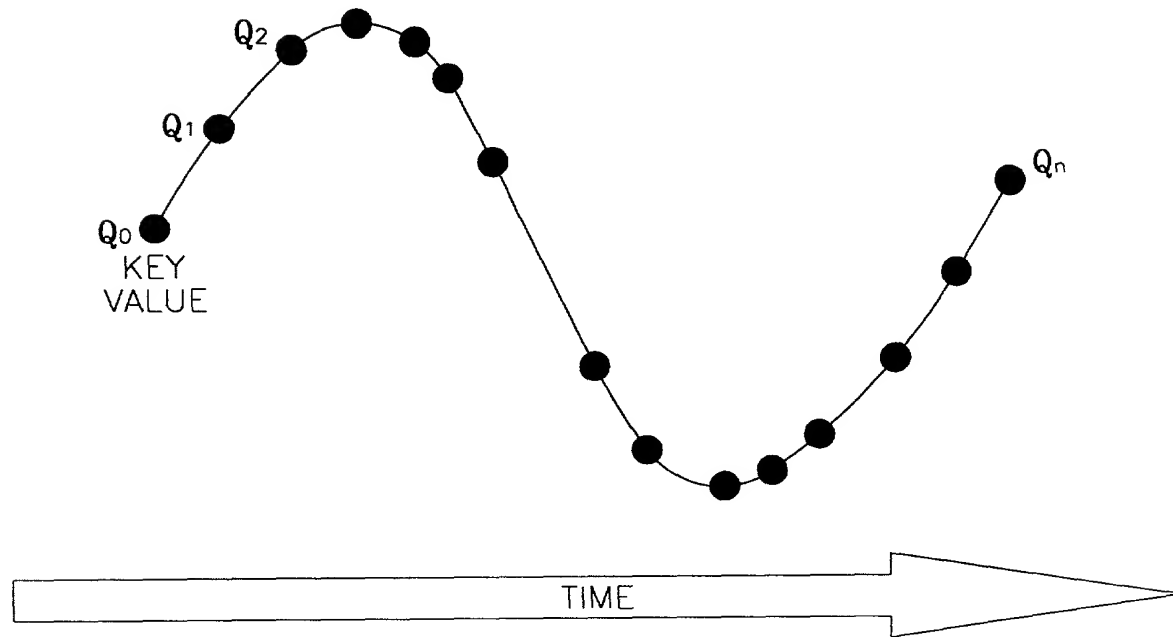


FIG. 14B

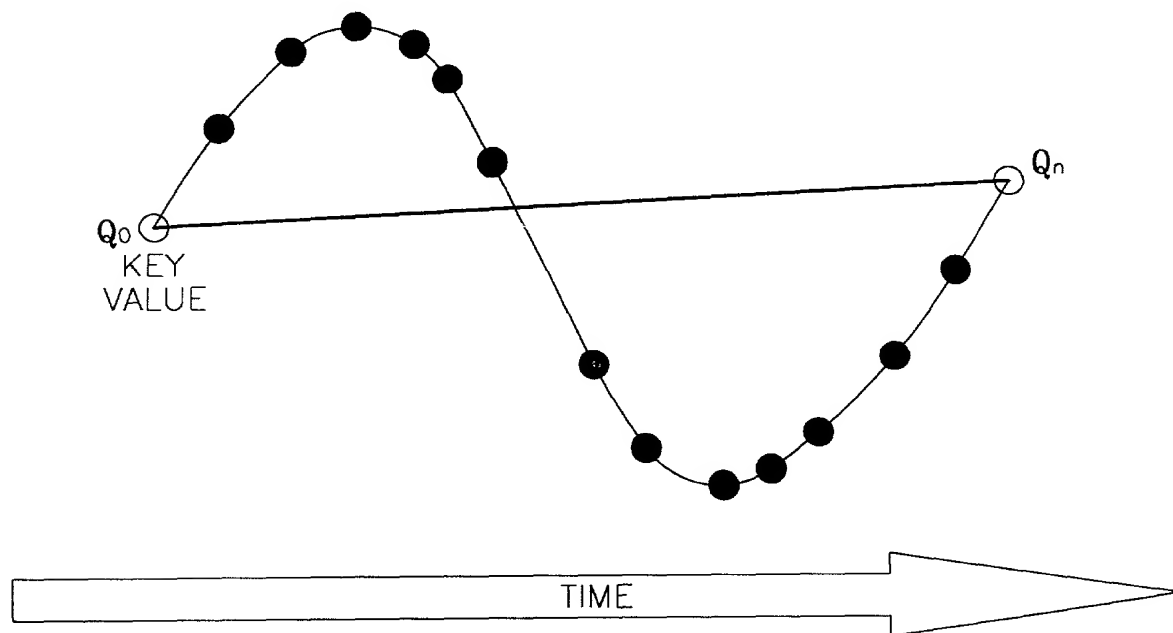


FIG. 14C

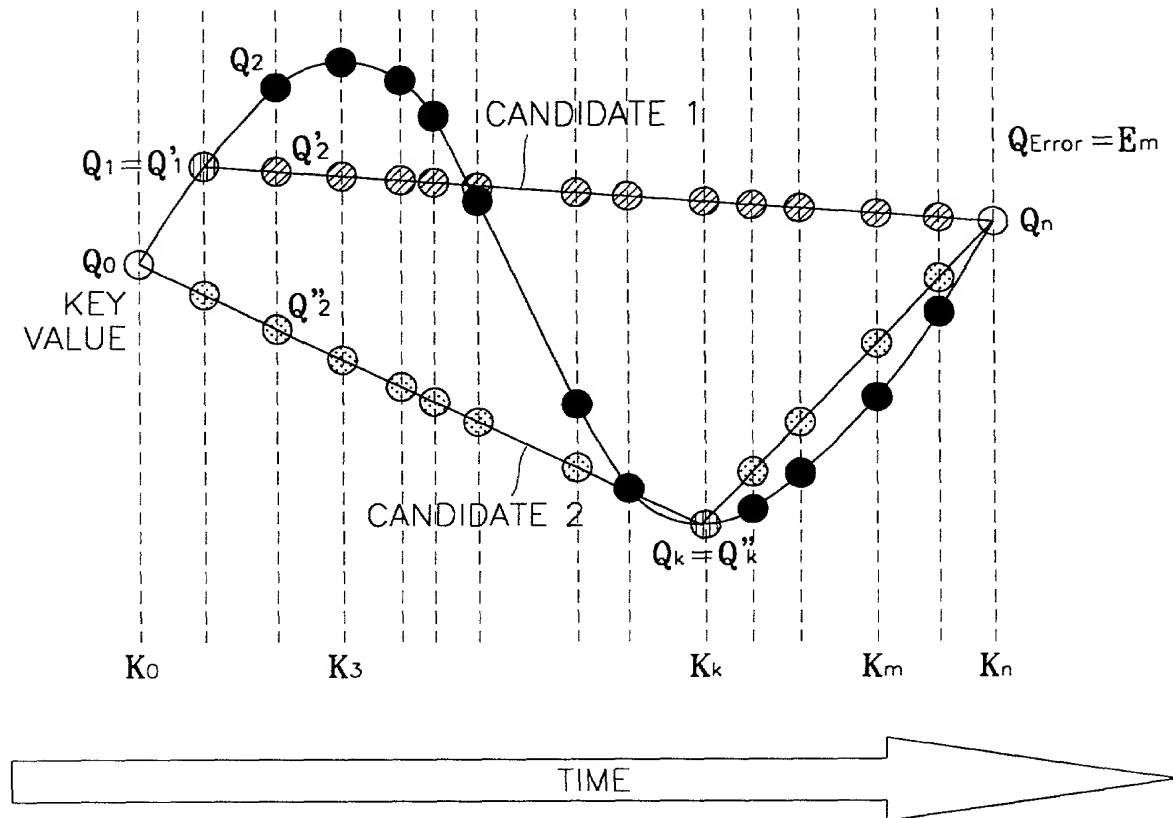


FIG. 14D

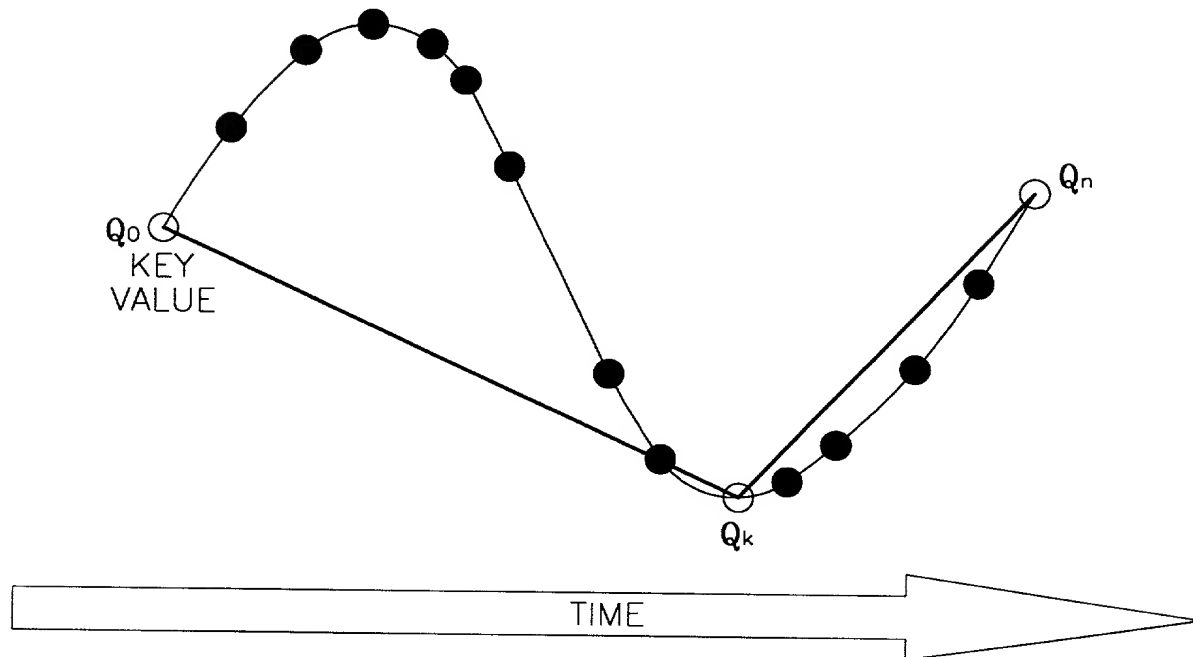


FIG. 14E

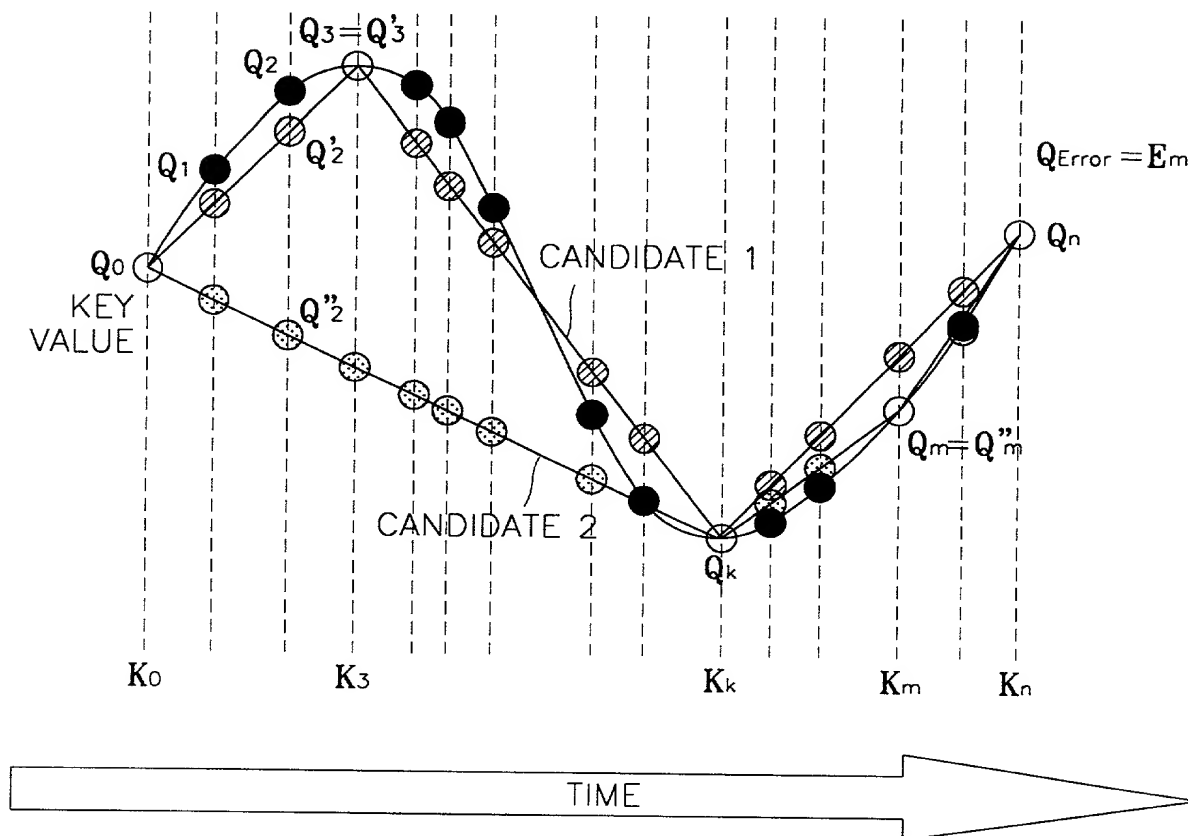


FIG. 14F

